A Neural Named Entity Recognition Approach to Biological Entity Identification

Emily Sheng, Scott Miller, José Luis Ambite, Prem Natarajan
Information Sciences Institute/USC, Marina del Rey, USA

***Abstract*—We approach the BioCreative VI Track 1 task of biological entity identification by focusing on named entity recognition (NER) and linking tagged entities to standard database identifiers. For this task, we apply recent neural NER techniques of combining bi-directional long short term memory (BLSTM) network layers with conditional random fields (CRFs) to the biomedical domain. We then use context words, dictionary lookups, and external biological knowledge bases to match tagged biological entities with corresponding identifiers. Our system predicts cell types and cell lines, cellular components, organisms and species, proteins and genes, small molecules, and tissues and organs.**

Keywords—named entity recognition; NER; bi-directional LSTM; conditional random fields; CRF; dictionary lookup

I. Introduction

In Track 1 of the BioCreative VI tasks, we are asked to automatically identify biological entities in biomedical text and link them to their standard database identifiers (e.g., UniProt ID for proteins and genes, ChEBI ID for small molecules, etc). Given biological entities annotated with identifiers in figure captions from 570 full-length articles as training data, we need to recognize and link biological entities in figure captions from 196 unseen articles to identifiers. Precision, recall, and F-measure at the caption, document, and corpus level for different entities are calculated. Additionally, there are also distinct scores for strict versus overlapping entity boundary matches, and for measuring across all annotations (i.e., including entities linked to a generic term such as "protein" and entities linked to a standard database identifier) versus just normalized entities (i.e., only entities linked to an identifier). It is useful to work with figure captions because sentences in captions describe figure objects, which are often the biological entities of interest. Also, extracting textual information from figure captions allows us to potentially link the textual data with figure data. Having methods to automatically extract and ground entities would be beneficial to the progression of research in scientific communities.

We train a CRF-based model [4] using NERSuite [3] as an NER baseline and compare it to the neural model. Although we are exploring different neural NER architectures, time constraints dictate that we use the LSTM-CRF architecture described in [5] as the NER model in our task submission. Our entity grounding component is based on dictionary and API lookups, and we apply some heuristics to more accurately segment entities for grounding.

This work was supported in part by the DARPA Big Mechanism program (W911NF-14-1-0364).

II. Preprocessing

We do as little preprocessing as possible to keep this component of the pipeline simple, generalizable, and easy to re-assemble into the BioC output format. For each caption paragraph, we use the NTLK (http://www.nltk.org/) sentence tokenizer to extract individual sentences, and then split each sentence on whitespace into "words" that we attach annotations to in the CoNLL format. Note that although we refer to the tokens divided by whitespace in a sentence as "words", these "words" can contain varying amounts of punctuation and are not necessarily well-formed English words. If the last byte in a sentence is ".", "?", or "!", we separate it into an additional "word". NERSuite takes a file with one sentence per line as input, while the neural NER model takes data in the CoNLL format as input. Both models output a tag for each "word" in the IOB format [7]. Our simple data tokenization into "words" and "word"-level tags means that there could potentially be multiple ground truth entities and entity types in one model-tagged entity or extraneous characters in model-tagged entities. If there are multiple annotations for a "word", we take the first annotation of an entity the word belongs to as the ground truth, and ignore all subsequent annotations that include this "word". We take sentences from a random 80% of the 570 articles to form the training set, sentences from 10% of the articles to form the development set, and sentences from the remaining 10% of the articles to form the test set.

TABLE I. Distribution of annotated entity types in training, development, and test sets

Entity type	Training set	Dev set	Test set
gene_or_protein	39,190 (54.41%)	4,312 (50.73%)	4,945 (57.71%)
small_molecule	8,602 (11.94%)	1,324 (15.58%)	857 (10.00%)
cellular_component	5,970 (8.29%)	530 (6.24%)	617 (7.20%)
cell_type_or_line	8,116 (11.27%)	966 (11.36%)	937 (10.94%)
tissue_or_organ	4,638 (6.44%)	538 (6.33%)	651 (7.60%)
organism_or_species	5,511 (7.65%)	830 (9.76%)	561 (6.55%)
Total	72,027 (100.00%)	8,500 (100.00%)	8,568 (100.00%)

Table 1 shows that a little more than half of the annotated entities across the training, development, and test sets are *gene_or_protein*.

III. Named Entity Recognition

A. NERSuite

We train a CRF-based baseline using NERSuite, so that we may compare the effects of our neural NER approach with a more standard CRF model. NERSuite is a toolkit that uses features derived from a tokenizer, part-of-speech tagger, lemmatizer, chunker, and optionally, dictionaries, as input into a CRF model. For our baseline model, we use all of the standard features except dictionaries to train NERSuite on our training and development set; we report results on the test set.

B. BLSTM-BLSTM-CRF

1) Related work: In recent years, a popular model for NER has been to derive character embeddings from a BLSTM or CNN model, combine the character embeddings with word embeddings and feed the concatenated result into another BLSTM layer. Some works additionally include a CRF layer that takes the output of the BLSTM layer as input. Chiu et al. [2] feed character embeddings and additional character features into a convolutional neural network (CNN) layer, and then concatenate the extracted character representation with word embeddings and additional word features to feed into a BLSTM layer. The BLSTM output is then forwarded to output layers to predict the best sequence of tags for a sentence. Lample et al. [5] concatenate word embeddings and BLSTM-extracted character embeddings to feed into a BLSTM layer, and then feed the BLSTM output to a CRF layer. Ma and Hovy [6] input character embeddings into a CNN layer, and then concatenate the extracted character representation with word embeddings to forward to a BLSTM and then CRF layer.

In our submitted model, we use the architecture shown in Fig. 1 and described in [5].

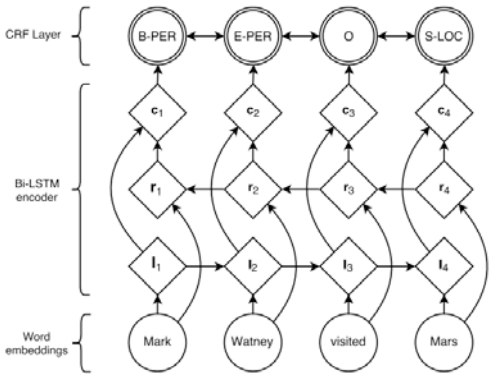

Fig. 1. This figure is taken from [5]. We apply their network with minor changes to the BioCreative dataset.

2) Word embeddings: Many previous works report that using pretrained word embeddings instead of randomly initializing word embeddings can significantly help increase NER scores. In our model, we use word embeddings pretrained on a combination of all abstracts from PubMed, all full-text from PubMed Central (a collection of open access documents from PubMed), and a Wikipedia dump [1]. We observe that these pretrained word embeddings boost the scores across all biological entity types significantly.

3) Character-based representation: To derive a character-based representation of words, we randomly initialize a 25-dimension vector for each character and input it to a BLSTM layer with a 25-dimension hidden layer. We assume, similarly to [5], that the hidden layer values of the last character in a word encode the character-level context of the entire word in a forward LSTM pass. Thus, we concatenate the hidden values of the last character in a word in the forward LSTM pass with the hidden values of the first character in a word in the backward LSTM pass to create the character-based representation of the word.

4) BLSTM-CRF with final word representation: To create an informative word representation, we concatenate the word embedding of a word with the character-based representation of the word. The final word representation is then input into a dropout layer, where half of the values from the word representation are dropped in training. Next, the word representations are input into a BLSTM layer with a hidden dimension of size 200. Similarly to the character-based BLSTM, we concatenate the hidden values of the last word in a sentence in the forward LSTM pass with the hidden values of the first word in a sentence in the backward LSTM pass to form the layer output. We then pass the output to a hidden layer to shrink the vector dimension back down to size 200, and use another hidden layer to shrink the vector to a dimension equal to the number of unique NER tags in the training data using the IOBES annotation scheme. In our training data, there are 25 such unique NER tags. The NER model internally uses the IOBES scheme, which also keeps track of singleton annotations and the end tokens of annotations; the final model output uses the IOB scheme. Lastly, we pass the 25-dimensional output vector to a CRF layer, where the CRF will use the BLSTM output vector and transition scores between a pair of tags to maximize the probability of the groundtruth tag sequence in training.

5) Parameters: The pretrained word embeddings are 200-dimensional vectors, so we use a 200-dimensional hidden layer in the word BLSTM. Lample et al. [5] use 100-dimensional vectors trained on news corpora and a 100-dimensional hidden layer in the word BLSTM. The first hidden layer uses a tanh activation function, and the second hidden layer uses a sigmoid activation function. We use stochastic gradient descent with a learning rate of 0.01 as the optimization function.

IV. Postprocessing

Before passing on tagged entities to be linked to identifiers, we strip all punctuation in the string "!\"#$%'()*+,-./:;<=>?@[\\]^_`{|}~" recursively from the beginning of the tagged entity. We also strip all punctuation in the string

[1] The details of the pretrained word embeddings are at http:/bio.nlplab.org/.

"!\"#$()*+,-./:;<=>?@[\\]^_`{|}~" from the end of the tagged entity. This does not completely remove all extraneous characters from tagged entities, nor does it remove only extraneous characters, but it works as a simple heuristic[2]. We attempt to find multiple proteins or genes that exist within a larger entity tagged as a protein or gene, but otherwise we do not further address the problems of multiple entities or entity types in a tagged entity in this work.

V. GROUNDING

A. Contextual dictionary

For annotated entities that are part of a larger word in the original sentence (e.g., "Tau" in "EcrTgTaumouse"), the words around the entity ("EcrTg" and "mouse" in the previous example) can be useful context words for linking to a standard identifier. We create a dictionary that maps all annotated entities that occur in the 570 training documents to the list of identifiers that entities have been linked to. Additionally, for each (entity, identifier) pair, we create a list of all the context "words" that are known to be associated. For example, for the entity "Tau", we could have the associated identifiers: "NCBI gene:17762" and "NCBI gene:4137". We further note that when "Tau" is linked to "NCBI gene:17762", the context words "EcrTg" and "mouse" are used. When "Tau" is linked to "NCBI:4137", the context words "EcrTg" and "human" are used. We do not weight context words based on how often they occur for an (entity, identifier) pair; this is left for future work. For each tagged entity that our NER model finds, we first check if the entity is in this contextual dictionary. If so, we link the entity to the known identifier that shares the most contextual words with the sentence the entity belongs to. Taking our example above, if the sentence were "Tau composition in ECrTgTau and control mouse lines investigated," the identifier "NCBI gene:17762" would have more context words in common with the words in the sentence and thus be assigned. Our intuition is that context words are strong indicators of species and other differentiating factors between identifiers of entities with the same surface forms.

B. Searching external knowledge bases

Table 2 shows the knowledge bases, API sources, and generic labels associated with each entity type. If we do not find a tagged entity in the contextual dictionary, we try to search for it in the appropriate external knowledge base. All searches for the submitted model were done on Aug. 22, 2017.

For proteins and genes, we use UniProt's official API[3] to search for entity identifiers. We use NCBI's Entrez tool to search the "taxonomy" database for identifiers for organisms and species [8]. For the rest of the entities, we use AmiGO, which is a collection of tools for searching the Gene Ontology database as well as knowledge bases for a few other ontologies [1]. We assign the first identifier match found in the appropriate knowledge base through the API source. If no matches are returned, and if the entity type is not *gene_or_protein*, we assign the entity to its generic label.

C. Further heuristics for proteins and genes

Proteins and genes have the most number of samples out of all the annotated biological entity types and also numerous variations in surface forms. For example, proteins "Tau", "MAPT", "MAPTL", and "MTBT1" are all synonyms. In contrast, the cellular component "ribosome" has synonyms "ribosomal RNA", "free ribosome", and "membrane bound ribosome", which are more similar to each other in surface form. We manually review some examples of proteins and genes in the training data and devise the following heuristic:

- If the complete tagged entity is not found through searching the UniProt API:
    - If there is whitespace in the tagged entity, split on the whitespace
    - Else if there is a forward slash "/" in the tagged entity, split on the "/"
    - Else if there is a dash "-" in the tagged entity, split on the "-"
    - Else if there is a semicolon ";" in the tagged entity, split on the ";"
- Search each split entity through the UniProt API. If identifiers are found, link the entity to the first found identifier. If no identifiers are found or if there are no whitespace, "/", "-", or ";" characters in the entire entity, just assign the generic label "protein".

TABLE II. SOURCES OF INFORMATION ACROSS ENTITY TYPES

Entity type	Knowledge base	API source	Generic labels
gene_or_protein	UniProt	UniProt	protein
small_molecule	ChEBI	AmiGO	molecule
cellular_component	GO	AmiGO	subcellular
cell_type_or_line	CL	AmiGO	cell
tissue_or_organ	Uberon	AmiGO	tissue
organism_or_species	NCBI taxon	Entrez (db: taxonomy)	organism

TABLE III. PRECISION, RECALL, AND F_1 SCORES ACROSS ENTITIES FOR DIFFERENT NER MODELS

Entity type	NERSuite			BLSTM-BLSTM-CRF		
	P	***R***	***F_1***	***P***	***R***	***F_1***
gene_or_protein	76.09	79.83	77.91	86.52	88.37	87.43
small_molecule	72.77	60.13	65.85	77.07	66.28	71.27
cellular_component	73.57	70.07	71.78	79.30	65.80	71.92
cell_type_or_line	67.60	62.59	65.00	76.85	65.53	70.74
tissue_or_organ	68.34	49.26	57.25	70.58	58.22	63.80
organism_or_species	61.89	65.08	63.44	72.59	75.04	73.79
overall	73.36	71.90	**72.62**	82.25	78.87	**80.53**

[2] These punctuation strings are the ones we used in the submitted model, but have since been revised.

[3] http://www.uniprot.org/help/api

TABLE IV. PRECISION, RECALL, AND F_1 SCORES ACROSS ENTITIES FOR SUBMITTED BIOLOGICAL ENTITY IDENTIFICATION MODEL

Entity types	Strict span match for all annotations			Strict span match for norm. annotations only			Span overlap match for all annotations			Span overlap match for norm. annotations only			Micro-averaged scores for normalized IDs			Macro-averaged scores across captions for normalized IDs		
	P	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***	***P***	***R***	***F1***
gene_or_protein	50.87	61.31	55.60	52.96	59.61	**56.09**	68.55	82.63	**74.94**	61.51	69.24	65.15	16.98	22.43	19.33	23.24	30.37	16.69
small_molecule	56.23	45.09	**50.04**	65.90	32.22	43.28	68.31	54.78	**60.80**	70.54	34.49	46.33	65.42	39.37	49.16	77.23	47.70	34.11
cellular_component	54.77	43.94	48.76	61.24	41.03	**49.14**	62.89	50.46	**55.99**	65.13	43.64	52.26	54.98	44.97	49.47	67.25	53.98	35.05
cell_type_or_line	65.31	65.03	65.17	82.23	55.15	**66.02**	76.63	76.30	**76.47**	86.58	58.06	69.51	78.42	55.69	65.13	85.58	60.15	55.51
tissue_or_organ	57.24	55.87	**56.55**	61.46	46.67	53.05	67.05	65.44	**66.24**	66.74	50.68	57.61	58.38	44.21	50.32	69.90	54.63	37.92
organism_or_species	74.62	71.52	73.04	85.52	69.50	**76.68**	81.36	77.98	**79.63**	87.82	71.37	78.74	77.23	69.10	72.94	82.48	74.69	65.46

VI. RESULTS AND CONCLUSIONS

A. NER results

Table 3 shows the results of the NERSuite model and the results of the BLSTM-BLSTM-CRF model. Both models were trained on the training and development sets and scores are reported for the test set. The BLSTM-BLSTM-CRF model using word embeddings pretrained on the biomedical domain does significantly better than the NERSuite model across all entity types. Our earlier experiments showed that for certain entity types, the BLSTM-BLSTM-CRF model with randomly initialized word embeddings performs comparably or worse than a CRF model when trained on the BioCreative data. This result emphasizes the significance of word embeddings pretrained on domain-specific data. We are in the process of experimenting with other neural architectures and are seeing promising results. Given that neural models often rely on a large amount of data to generate accurate results, and biomedical NER annotations need to be done by experts, we think distant supervision techniques may be especially helpful. Our manual error analysis indicates that better tokenization schemes might help the model better detect the boundaries of an entity. Also, our model tags entities at the word level, but the task evaluates entities at the byte level, so a model that tags at the byte level may be more suitable.

B. Submission results

In Table 4, we list the scores under various evaluation conditions for the unseen test set. The first four conditions evaluate the NER model, and the last two conditions evaluate grounding tagged entities to identifiers. As expected, the scores are higher when evaluating span overlap entity matches versus strict span entity matches. Interestingly, all entity types have higher F_1 scores when evaluating span overlap match for all annotations versus for normalized annotations only. This indicates that our NER model is better at detecting non-normalized entities across entity types.

In this work, we focus on experimenting with state-of-the-art NER techniques applied to the biomedical domain. We do not spend a comparable effort on grounding techniques, though we are working to improve them. We observe that the grounding method in our submitted model performs the best on *organism_or_species* and *cell_type_or_line*, does ok on *small_molecule*, *cellular_component*, and *tissue_or_organ*, and performs poorly on *gene_or_protein*. One reason the normalization performance of *gene_or_protein* entities is poor is because we use a limited context to ground entities. Similar genes and proteins of difference species often have the same surface forms, and the only way to accurately ground the genes and proteins is to infer the species from the textual context. Another explanation for the poor normalization performance is that genes and proteins have the most variations in surface forms; there are relatively fewer ways to refer to organisms and species, for example. From a manual evaluation of our grounding method, we observe that better organism modeling would help improve the normalization scores. Also, the simple heuristics for segmentation seem to help us more accurately extract short protein and gene entities, but we often make more errors grounding shorter proteins and genes. For future work, we would explore using more contextual evidence to assign entity identifiers.

REFERENCES

[1] Carbon S, Ireland A, Mungall CJ, Shu S, Marshall B, Lewis S, AmiGO Hub, Web Presence Working Group. AmiGO: online access to ontology and annotation data. Bioinformatics. Jan 2009;25[2]:288-9.

[2] Chiu, J. P., & Nichols, E. (2015). Named entity recognition with bidirectional LSTM-CNNs. *arXiv preprint arXiv:1511.08308*.

[3] Cho, H. C., Okazaki, N., Miwa, M., & Tsujii, J. (2010). NERsuite: a named entity recognition toolkit. *Tsujii Laboratory, Department of Information Science, University of Tokyo, Tokyo, Japan*.

[4] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[5] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

[6] Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

[7] Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora* (pp. 157-176). Springer Netherlands.

[8] Sayers E. E-utilities Quick Start. 2008 Dec 12 [Updated 2013 Aug 9]. In: Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK25500/